

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
df=pd.read_csv("/home/student/Desktop/Iris.csv")
df
```

```
Out[1]:
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

```
In [2]: df.isnull().any()
```

```
Out[2]: Id                False
SepalLengthCm           False
SepalWidthCm            False
PetalLengthCm           False
PetalWidthCm            False
Species                 False
dtype: bool
```

```
In [3]: x=df.iloc[:, :4].values
y=df['Species'].values
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.2,random_sta
```

```
In [4]: from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
xtrain=scaler.fit_transform(xtrain)
xtest= scaler.transform(xtest)
```

```
In [5]: from sklearn.naive_bayes import GaussianNB
gaussian=GaussianNB()
gaussian.fit(xtrain,ytrain)
y_pred=gaussian.predict(xtest)
y_pred
```

```
Out[5]: array(['Iris-virginica', 'Iris-versicolor', 'Iris-setosa',
               'Iris-virginica', 'Iris-setosa', 'Iris-virginica', 'Iris-setosa',
               'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
               'Iris-virginica', 'Iris-versicolor', 'Iris-versicolor',
```

```
        'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',  
        'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa', 'Iris-setos  
a',  
        'Iris-virginica', 'Iris-versicolor', 'Iris-setosa', 'Iris-setosa  
,  
        'Iris-virginica', 'Iris-setosa', 'Iris-setosa', 'Iris-versicolor  
,  
        'Iris-versicolor', 'Iris-setosa'], dtype='<U15')
```

```
In [6]: from sklearn.metrics import precision_score, confusion_matrix, accuracy_s  
cm=confusion_matrix(ytest, y_pred)  
cm
```

```
Out[6]: array([[11,  0,  0],  
               [ 0, 13,  0],  
               [ 0,  0,  6]])
```

```
In [7]: accuracy=accuracy_score(ytest, y_pred)  
accuracy
```

```
Out[7]: 1.0
```

```
In [8]: precision=precision_score(ytest, y_pred, average="micro")  
precision
```

```
Out[8]: 1.0
```

```
In [9]: recall=recall_score(ytest, y_pred, average="micro")  
recall
```

```
Out[9]: 1.0
```

```
In [ ]:
```